

NASA Technical Memorandum 89109

A COMPUTER SIMULATOR FOR DEVELOPMENT OF
ENGINEERING SYSTEM DESIGN METHODOLOGIES

S. L. Padula and J. Sobieszczanski-Sobieski

(NASA-TM-89109) A COMPUTER SIMULATOR FOR
DEVELOPMENT OF ENGINEERING SYSTEM DESIGN
METHODOLOGIES (NASA) 18 P CSCL 12A

N87-20755

G3/59 45261
Unclas

February 1987



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

A COMPUTER SIMULATOR FOR DEVELOPMENT OF ENGINEERING SYSTEM DESIGN METHODOLOGIES

S.L. Padula, and J. Sobieszczanski-Sobieski

SUMMARY

The simulator is a computer program which mimics the qualitative behavior and data couplings occurring among the subsystems of a complex engineering system. It eliminates the engineering analyses in the subsystems by replacing them with judiciously chosen analytical functions. With the cost of analysis eliminated, the simulator is used for experimentation with a large variety of candidate algorithms for multilevel design optimization methodologies to choose the best ones for the actual application. Thus, the simulator serves as a development tool for multilevel design optimization strategy. The simulator concept, implementation, and status are described and illustrated with examples.

INTRODUCTION

Complex engineering systems, e.g., an aircraft, a car, a ship, or an electric power station are usually amenable to decomposition in which the whole is treated as an assembly of smaller parts. Traditionally, engineers have used this technique to break their large design task into smaller subtasks executed concurrently, thus developing a broad workfront and compressing the design process schedule. Recently, that approach has been augmented by numerous formal methods incorporating mathematics into what used to be a predominately heuristic practice, e.g, (refs. 1 and 2). The simulator to be described in this paper is a tool for the development of such methods.

Schematically, the decomposition may be presented as a pyramid of hierarchically related modules, each corresponding to a design subtask. The subtasks may correlate with physical subsystems or with engineering disciplines contributing to the system design. In the former case, the decomposition is called an object decomposition, while the latter case is known as an aspect decomposition (ref. 1).

The way the content of each module (box in fig. 1) is defined depends on the intended use of the diagram. For management purposes the modules are groups of people. For the purposes of this paper each module represents an algorithm converting input into output. The algorithm may include both analysis and optimization. Consistent with that definition, the directed lines in fig. 1 portray information flow (data channels) from one module to another.

Obviously, execution of a design process organized in a manner depicted in fig. 1 requires specification of all the module algorithms and definition of the meaning and volume of the data moving along each channel. It also requires definition of an overall procedure sequencing the module algorithms

in time. Here, the overall procedure takes the form of a multilevel optimization whose purpose is to satisfy constraints and improve the performance of the whole system.

In the present paper, the module algorithms will be treated as black boxes defined to the remainder of the system by their input-to-output transformation, and the input-output data content. This assumption leaves us free to concentrate on the issues of the data exchange among the modules, and the effective and efficient organization of an iterative algorithm for the multilevel optimization of the decomposed system. The simulator program described in this paper is a tool for doing that without paying the labor and computer costs of analyses that would have to be repetitively carried out inside the modules of the decomposed system if a real physical engineering system was used as a case study. From a research and development standpoint these costs can be prohibitive.

The basic idea which makes the simulator operate inexpensively is to replace the detailed engineering analysis in each module by explicit, simple functions that model qualitatively the module behavior. The paper's purpose is to describe that idea in detail, with enough basic information about multilevel optimization to put the simulator in context of the methodology development. The discussion will include information on the computer implementation of the simulator, its development status, and a review of typical results obtained to date. Familiarity with the concept of design optimization by search for a constrained optimum in a design space, and with the pertinent terminology is assumed.

DECOMPOSITION AND MULTILEVEL OPTIMIZATION

Multilevel optimization relies on object or aspect decomposition of a system to break the system optimization task into a set of suboptimization tasks and a coordination task which restores the coupling among the subtasks. It can be best explained by contrasting it with a conventional optimization without decomposition.

In a conventional optimization we define for the entire system a vector of design variables X , and a set of inequality constraints $G(X)$. No distinction is made among the elements of X , and G that may belong to different subsystems. Choosing a suitable system performance measure as the system objective function $F(X)$, we solve a classical optimization problem

$$\min_X F(X) \text{ subject to constraints } G(X) \leq 0; \quad L_X \leq X \leq U_X \quad (1)$$

where L_X , U_X are side constraints, using a nonlinear mathematical programming (NLP) procedure starting with a "best guess" initial X . The numerical information about the values of the functions F , G and their gradients needed by the NLP procedure comes from the analyses of the system modules and from the analysis of the assembled system. This implies that the system may still be decomposed for the purpose of analysis but not for the purpose of optimization.

In a multilevel optimization, the system is decomposed for both the analysis and optimization purposes as shown in fig. 1. The system symbolized by the

box on the top of the pyramid (level 1) is a "parent" to the "daughter" subsystems at level 2. The parent's output becomes the daughter's input, PI. Each daughter may be a parent to daughters at the next lower level - a recursive relationship that may extend to unlimited number of levels.

For each subsystem we define: Y - a subset of X ; g - a subset of G ; and C - a "local" objective function. For the assembled system we define also an objective function $F(X_s)$ and a subset of constraints $g_s(X_s)$, where X_s represent the system design variables.

Several methods exist for optimization of a decomposed system. The algorithm introduced in (refs. 2 and 3) will be used here as an example by which to introduce the simulator and to describe its mechanism. That algorithm can be summarized by the following steps:

1. Initialize all design variables.
2. Analyze from the top down.
3. Optimize each subsystem proceeding from the bottom up (concurrent suboptimization tasks).
4. Optimize the assembled system (the coordination task).

In step 2, output from each parent is used as input in the daughter analyses. In step 3, the optimization problem solved for each subsystem separately, beginning at the very bottom of the hierarchy, is

$$\min_Y C(Y) \text{ subject to constraints } h = PI(X) - f(Y) = 0 \quad (2)$$

$$L_Y \leq Y \leq U_Y$$

where L_Y and U_Y are the side constraints.

The "local" objective function C is formulated as a cumulative constraint (ref. 4), written in form of a function

$$C = \frac{1}{\rho} \ln\left(\sum_j \exp(\rho g_j)\right) \quad (3)$$

where ρ is a constraint satisfaction tolerance. The function defined by eq. 3 (introduced in (ref. 5)) has the property of being a differentiable approximation to the maximum constraint in a particular subsystem, so that

$$\max(g) \leq C \leq \max(g) + \ln(m)/\rho \quad (4)$$

where m is the number of the constraint functions.

The equality constraints h in eq. 2 are needed whenever some of the elements of PI are functions not only of X but also of Y . For example, if a subsystem is a beam member in a framework structure, then its cross-sectional area may be imposed on it from above as an element of PI . However, that area is also a function of the beam cross-sectional dimensions Y . In such cases, the element of PI prescribes a certain property of the subsystem and the constraints h enforce that prescription by equating the PI to some function

of the daughter variables Y . In effect, any changes in the local design variables, Y , are restricted so as to maintain the cross-sectional area constant. Owing to the constraints h , the validity of all the PI vectors obtained in step 2 is protected when the subsystems are optimized in step 3. The functions of X and Y that appear together in an equality constraint h will be referred to as the coupling functions. They represent one particular conduit for a parent to influence its daughter. There are other conduits for that influence that will be defined later.

Solution of eq. 2 yields a constrained minimum of C , denoted \bar{C} and the corresponding solution vector \bar{Y} . Derivatives of \bar{C} with respect to the elements of PI are now calculated using the optimum sensitivity algorithm introduced in (ref. 6), to obtain an approximate value of \bar{C} , denoted \tilde{C} , as a function of PI. That function, expressed by the linear part of the Taylor series, is

$$\tilde{C} = \bar{C} + \sum_k \frac{\partial \bar{C}}{\partial PI_k} \Delta PI_k; \quad k = 1 \rightarrow z; \quad (5)$$

where z is the number of the parent inputs to the subsystem. Since PI is a function of X , \tilde{C} is ultimately related to X

$$\tilde{C} = \bar{C} + \sum_k \sum_q \frac{\partial \bar{C}}{\partial PI_k} \frac{\partial PI_k}{\partial X_q} \Delta X_q \quad (6)$$

where q identifies the elements of X that influence the PI.

Proceeding, still within step 3, to the next level up, the daughter \bar{C} approximated by \tilde{C} in eq. 6 is appended to the parent vector g as another inequality constraint. That means the information about the subsystem constraint satisfaction, or violation, measured by the \tilde{C} quantities accumulates recursively, and in step 4 the system optimization problem becomes:

$$\min_{X_s} F(X_s) \text{ subject to constraints } g_s(X_s) \leq 0 \quad (7)$$

$$\tilde{C}_i \leq 0; \quad i = 1 \rightarrow r$$

where r is the number of subsystems in level 2.

The procedure is iterated in order to update the analysis and sensitivity information according to the changed values of the design variables, until all the constraints are satisfied at all levels and the system objective function converges.

In the above discussion it was assumed that the data flow in the system analysis (step 2) is strictly top down, and that each daughter has only one parent. This is a simplification. In decomposition of the real engineering systems, the data flow pattern may be more complicated as illustrated in fig. 2. In addition to the top-down, single parent flow, we may have:

1. more than one parent per daughter (multiple parents);
2. output from a daughter needed as input into the parent analysis (reverse interaction);
3. output from a daughter needed as input into another daughter (sister)

analysis at the same level (lateral interaction); 4. input/output channels extending beyond the next lower, or higher, level (multilevel span), which impacts the simulator implementation and its data management. An extensive study of a transport aircraft as an engineering system (ref. 7) provided numerous examples of such complexities.

The above complications in the data flow pattern have to be properly reflected in the optimization algorithm. Specific algorithm augmentations designed to handle the situations 1 through 4 were proposed in (ref. 3). However, no rigorous, mathematical means could be identified to ascertain the convergence properties of the algorithm whether in the simplest form introduced in the foregoing discussion, or with characteristics 1 through 4 above. This amplifies the importance of numerical testing of multilevel design optimization algorithms for convergence and other performance characteristics.

THE SIMULATOR CONCEPT

The algorithm in its simplest form has been tested with good results in applications to structural optimization (refs. 4, 8, and 9), and to multidisciplinary problems in aeronautics (refs. 10 and 11). While validation by engineering system test cases is a necessary part of the methodology development, the referenced experience showed that in such testing the cost of subsystem analysis is so large that it severely restricts the scope of experimentation that can be accomplished within given resources.

The simulator described in the remainder of this report is intended to be a means by which an exhaustive experimental testing of the multilevel optimization algorithms can be conducted without paying the costs of detailed engineering analyses. The key to an inexpensive simulator is a replacement of the physical analyses in the subsystems by explicit analytical functions whose evaluation cost is negligible. Such functions can be constructed taking advantage of the insight into the behavior of typical physical subsystems. That behavior may be analytically complex, but quite frequently it is also descriptively simple and qualitatively well known in advance. For example, it may take a large finite element model analysis to determine the axial stress in a structural member, but it is well known that that stress will be diminishing with the increase of the member cross-sectional area, so that a simple, explicit function, $\text{stress} = \text{constant}/\text{area}$ captures that behavior.

It may be argued that the family of monotonic polynomials is adequate to represent a large subset of the objective functions and constraints encountered in engineering applications. With the increase of the design variables, these polynomials either increase, or decrease, with or without diminishing returns. Table 1 defines the functions, their nature, design variables, and parameters which are currently defined in the simulator. Function 1 is used only for the system objective functions therefore its variables are X_s . Functions 2 through 4 are used for the subsystem constraints so their variables are Y .

The coefficients of the polynomials are either randomly generated constants or they may be used as another conduit, in addition to the previously defined coupling functions, to transmit influence of one subsystem on another. Two mechanisms for generating that influence are shown in figs. 3 and 4.

The simpler mechanism depicted in fig. 3 substitutes a parent design variable for a parameter in the daughter analysis. The other way shown in fig. 4 introduces another type of a coupling function, denoted Q , computed in the parent analysis and substituted for a parameter in the daughter analysis. A structural system example of the Q -type of a coupling function is the boundary interaction force acting on a substructure. That force is computed in the analysis of the assembled structure and is considered as a constant load in the substructural analysis.

Both above coupling mechanisms may be used simultaneously. The coupling strength depends on the relative magnitude of the parameters transmitted and on the power to which they are raised in the daughter analysis.

The pattern in which the parameters are transmitted from one subsystem to another constitutes the means by which a variety of hierarchical relationships may be simulated, ranging from the simple top-down hierarchy shown in fig. 1 to a complex one described in fig. 2 and associated discussion.

The simulator implementation has been progressing from the simplest system toward increasing complexity and has reached the status summarized in table 2. For generating benchmark results the simulator provides an option of single level optimization in which the system analysis is decomposed but the optimization is defined by eq. 1. For the multilevel optimization purposes, the current implementation includes all of the function types defined in table 1, both types of the parent-daughter influences shown in figs. 3 and 4, and more than one parent per daughter case. The latter required an augmentation to the previously described multilevel optimization procedure, by introducing a cumulative constraint representing the minimized cumulative constraints of all the p subsystems at a given level

$$C_1 = \frac{1}{p} \ln \left(\sum_i \exp(\rho \bar{C}_i) \right); \quad i = 1 \rightarrow p \quad (8)$$

This constraint, approximated using the optimum sensitivity derivatives as in eq. 6, is appended to the vector of constraints in each of the subsystems at the next higher level. This allows the multiple parents to exert cross-influence on the shared daughters, in proportion to the magnitude of the corresponding optimum sensitivity derivatives.

All the optimizations in the subsystems and at the system level are carried out using the technique of usable-feasible directions implemented as described in ref. 12. The simulator has been coded as a modular Fortran program. The decomposed system is described by a data structure made up of the polynomial coefficients (table 1). In this study, some of the coefficients in that structure are arbitrary constants, e.g., all $r_i = 1$, others are randomly generated, e.g., the coefficients a , b , c , and d , and, finally, some coefficients are reserved to implement the coupling shown in fig. 3, e.g., the coefficients e^* and h^* . With the exception of those marked with the asterisk, the coefficients remain constant in the optimization execution and may be saved and used repeatedly. Details of the computer implementation are given in ref. 13. The simulator has also been implemented in a distributed manner on a network of computers to investigate benefits from concurrent execution of the subsystem analyses and optimizations.

SAMPLE OF RESULTS

Table 3 defines a simple system in which there is one parent per daughter and the mechanism shown in fig. 3 is used to substitute the parameters in the subsystems. This test case was used to compare the convergence of single and multilevel optimization methods. Figs. 5 and 6 compare the single level optimization history (histogram) with the multilevel optimization histogram for the same system. The system level objective is plotted versus the total number of constraint evaluations that are required as the iteration advances. The number of evaluations is taken as a measure of the computational cost, assuming that all the constraint functions are equally expensive to compute - an assumption approximately valid for the simulator, but not necessarily valid for engineering systems in general.

Fig. 5 depicts histograms for a particular initialization deliberately chosen to be quite close to the optimum. In this case, the multilevel optimization converges smoothly. On the other hand, the single level optimization converges quickly at first and then slows down nearing the optimum. After 800 constraint function evaluations, each method has identified a feasible solution. The solution reached by the multilevel optimization has a slightly lower objective.

Fig. 6 is a histogram of an optimization whose initialization is far from the optimum. Again, the multilevel optimization converges smoothly but this time it requires about 1400 function evaluations to identify a feasible solution and 200 additional evaluations to reach a final solution. In contrast, the single level optimization finds a feasible solution after only 800 function evaluations but then requires additional 600 function evaluations (1400 total) to reduce the objective to its final value.

To investigate the effects of the system coupling on the convergence, the complexity of the system defined in table 3 was increased by using the substitution pattern given in table 4. The effect of the revision is that each daughter has multiple parents. Fig. 7 and fig. 8 show the multiple parent effect on the single level optimization to be much stronger than the effect on the multilevel optimization. In both figures the multilevel optimizations are monotonic although somewhat slowed down in their convergence, while for the single-level optimization fig. 7 shows a jagged graph with an exceedingly slow terminal convergence phase and fig. 8 shows a failure to converge.

It should be pointed out, however, that at least part of the advantage of the multilevel optimization shown in the above comparisons may be attributed to the use in that method of the usable-feasible directions algorithm enhanced (at the system level only) with the well known constraint relaxation (ref. 14), while no such enhancement was implemented in the single level optimization.

CONCLUDING REMARKS

A simulator for multilevel optimization of complex hierarchical systems has been developed. Its purpose of radically reducing the analysis cost in experimentation with various multilevel design optimization algorithms was achieved by using explicit functions instead of computationally expensive analyses that would have to be executed in each subsystem, and choosing

these functions so as to preserve the subsystem response qualitatively. With the cost of analysis practically eliminated, the simulator can be used to investigate a wide range of multilevel optimization algorithms and system configurations.

The simulator demonstrated its usefulness as means for evaluating efficiency and effectiveness of the multilevel optimization algorithms, and revealing the effects of the subsystem couplings on their convergence characteristics. The experience to date showed for the cases tested: 1. Agreement of the multilevel optimization results with the benchmark results produced without decomposition; 2. Acceptable rate of convergence for the multilevel optimization algorithms tested, including instances where the multilevel optimization converged faster than the reference single level optimization; 3. The multilevel optimization rate of convergence slightly reduced when the strength and complexity of couplings was increased; moreover, there was no appreciable detrimental effect on the minimum of the system objective and ability to satisfy the constraints.

In summary, the simulator confirmed the viability of those multilevel optimization algorithms that were tested, and has been shown to be a useful tool in the development of these algorithms for the use in design of complex engineering systems.

REFERENCES

1. Archer, B.: The Implications for the Study of Design Methods of Recent Developments in Neighbouring Disciplines. Proceedings of International Conference on Engineering Design, ICED 1985, Hamburg, West Germany, Aug. 1985.
2. Sobieszczanski-Sobieski, J., and Barthelemy, J.-F. M.: Improving Engineering System Design by Formal Decomposition, Sensitivity Analysis, and Optimization. Proceedings of International Conference on Engineering Design, ICED 1985, Hamburg, West Germany, Aug. 1985.
3. Sobieszczanski-Sobieski, J.: A Linear Decomposition Method for Large Optimization Problems - A Blueprint for Development, NASA TM-83248, Feb. 1982.
4. Rogan, J.E., Kolb, M.A., and McElveen, R.T.: Application of Decomposition Techniques to Preliminary Design of Transport Aircraft. AIAA Paper No. 86-2617, AIAA Aircraft Systems Design and Technology Meeting, Dayton, OH, Oct. 1986, also pending NASA CR.
5. Sobieszczanski-Sobieski, J., James, B.B., and Dovi, A.R.: Structural Optimization by Multilevel Decomposition. AIAA J, Vol. 23, No. 11, p. 1775, Nov. 1985.
6. Kreisselmeier, G., and Steinhauser, G.: Systematic Control Design by Optimizing a Vector Performance Index. Proceedings of IFAC Symposium on Computer Aided-Design of Control Systems, Zurich, Switzerland, 1979.
7. Sobieszczanski-Sobieski, J., Barthelemy, J.-F. M., and Riley, M.F.: Sensitivity of Optimum Solutions to Problem Parameters. AIAA J, Vol. 20, No. 9, Sept. 1982, p. 1291.

8. Sobieszczanski-Sobieski, J., James, B.B., and Riley, M.F.: Structural Optimization by Generalized, Multilevel Decomposition. AIAA Paper No. 5-0697-CP, AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics and Materials Conference, Orlando, FL, April 1985.
9. Barthelemy, J.-F. M., and Riley, M. F.: An Improved Multilevel Optimization Approach for the Design of Complex Engineering Systems. AIAA Paper No. 6-0950-CP, AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics, and Materials Conference, San Antonio ,TX, May 1986.
10. Barthelemy, J.-F. M.: Development of a Multilevel Optimization Approach to the Design of Modern Engineering Systems. NASA CR-172184, Aug. 1983.
11. Wrenn, G.A., and Dovi, A.R.: Multilevel/Multidisciplinary Optimization Scheme for Sizing a Transport Aircraft Wing, AIAA Paper No. 87-0714-CP, AIAA/ASME/ASCE/AHS 28th Structures, Structural Dynamics, and Materials Conference, Monterey, CA, April 1987.
12. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design. McGraw-Hill, 1984.
13. Padula, S.L., and Young, K.C.: Simulator for Multilevel Optimization Research. NASA TM-87751, June 1986.
14. Barthelemy, J.-F. M., Chang, K.J., and Rogers, Jr., J.L.: Shuttle Rocket Booster Bolted Field Joint Optimization, Appendix. AIAA Paper No. 87-0702-CP, AIAA/ASME/ASCE/AHS 28th Structures, Structural Dynamics, and Materials Conference, Monterey, CA, April 1987.

Table 1.- The Simulator Functions

Nature	Expression	Variables	Parameters [†]
Increasing quadratically	$f_1 = \sum_i r_i x_i^2$	$x_i \ x_s$	r
Decreasing with diminishing returns	$f_2 = -a - \sum_i b_i y_i + \sum_i c_i y_i^{-1} + \sum_i d_k (e_k^* h_k^*)^{-1}$	$y_i \ Y$	a b
Increasing	$f_3 = -a + \sum_i b_i y_i + \sum_k d_k (e_k^* h_k^*)$		c d
Increasing with diminishing returns	$f_4 = -a + \sum_i b_i y_i^{1/2} + \sum_k d_k (e_k^* h_k^*)^{1/2}$		e h

† - positive real numbers.

* - equated to parent x.

Table 2.- Simulator Status

Characteristic	Implementation status
Function types	f_1, f_2, f_3, f_4 defined in table 1
Type of the parent-daughter influence	1. Coupling functions in the equality constraints h in eq. 2 - no 2. Mechanism illustrated in fig. 3 - yes 3. Mechanism illustrated in fig. 4, with the parameters e^* and h^* as the only ones subjected to substitution - yes
Complexity of the analysis data flow	
0. Single parent, top-down	yes
1. Multiple parents	yes
2. Reverse interaction	no
3. Lateral interaction	no
4. Multilevel span	yes
Multilevel optimization algorithm	0. Reference algorithm defined by eq. 1 1. Algorithm defined by eqs. 2-7 2. As above modified according to eq. 8
Search for constrained minimum	Usable-feasible directions technique at all levels

Table 3.- Four-Level System

<div>Objective<div>$\sum_1^3 x_i^2$</div></div> <div>Variables<div>x_1, x_2, x_3</div></div> <div>Parameters<div>None</div></div>				
No. of constraints	3	2	2	
Variables	y_1, y_2, y_3	y_6, y_7, y_8	$y_{12}, y_{13}, y_{14}, y_{15}$	
Parameters	x_1, x_2, x_3	x_1, x_2, x_3	x_1, x_2, x_3	
No. of constraints	1	2	3	2
Variables	y_4, y_5	y_9, y_{10}, y_{11}	y_{16}, y_{17}, y_{18}	y_{19}, y_{20}
Parameters	y_1, y_2, y_3	y_6, y_7, y_8	y_{12}, y_{13}	y_{14}, y_{15}
No. of constraints			2	
Variables			y_{21}, y_{22}	
Parameters			y_{16}, y_{17}, y_{18}	

1. Each box represents a subsystem - a daughter to a parent immediately above
2. Cumulative constraint C is used as the objective function in each subsystem optimization.

Table 4.- Four-Level System With Multiple Parent Couplings

<div>Objective<div>$\sum_1^3 X_i^2$</div></div> <div>Variables<div>X_1, X_2, X_3</div></div> <div>Parameters<div>None</div></div>				
No. of constraints	3	2	2	
Variables	$Y_1, Y_2, \underline{Y_3}$	$Y_6, \underline{Y_7}, Y_8$	$Y_{12}, \underline{Y_{13}}, Y_{14}, Y_{15}$	
Parameters	X_1, X_2, X_3	X_1, X_2, X_3	X_1, X_2, X_3	
No. of constraints	1	2	3	2
Variables	Y_4, Y_5	Y_9, Y_{10}, Y_{11}	Y_{16}, Y_{17}, Y_{18}	Y_{19}, Y_{20}
Parameters	$Y_1, Y_2, \underline{Y_3}, \underline{Y_7}, \underline{Y_{13}}$	$Y_6, \underline{Y_7}, Y_8, \underline{Y_3}$	$Y_{12}, \underline{Y_{13}}, \underline{Y_3}$	$Y_{14}, Y_{15}, \underline{Y_7}$
		No. of constraints	2	
		Variables	Y_{21}, Y_{22}	
		Parameters	Y_{16}, Y_{17}, Y_{18}	

1. The variables and parameters underlined with "-" are coupled in a multiple parent pattern.
2. Cumulative constraint C is used as the objective function in each subsystem optimization.

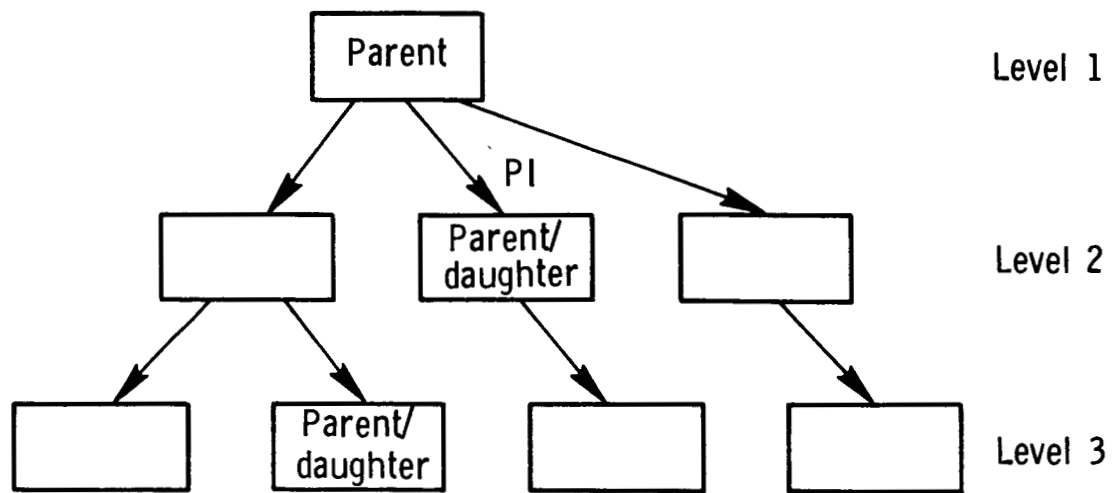


Fig. 1 Decomposed system: simple top-down case of the analysis data flow.

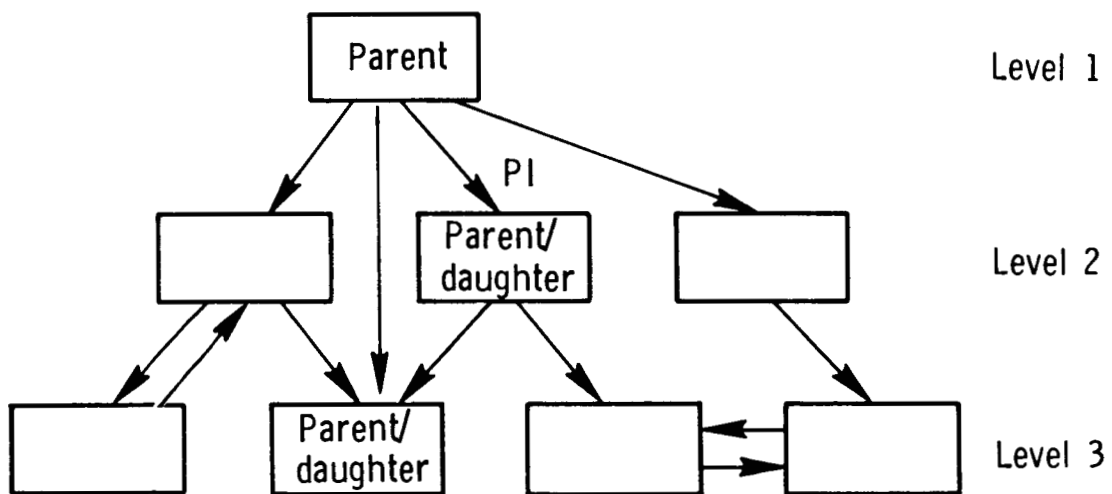


Fig. 2 Decomposed system with more complex data flow.

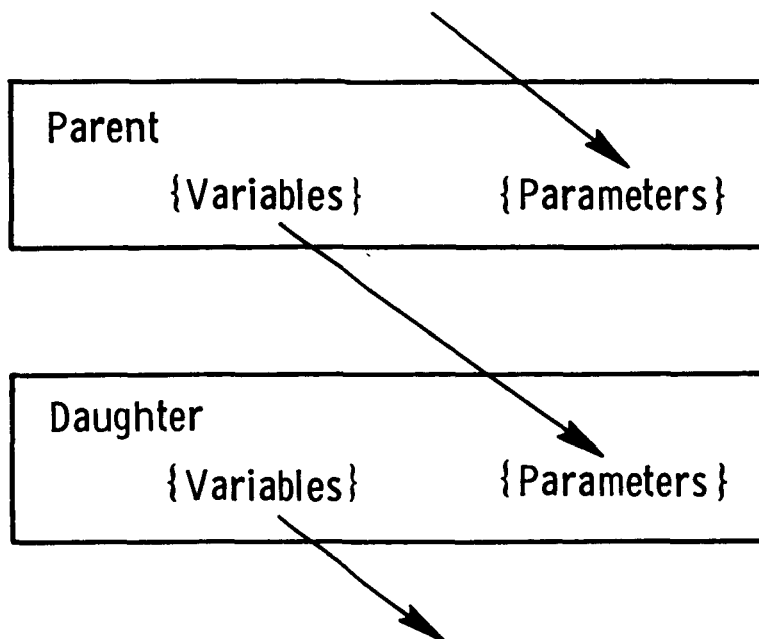


Fig. 3 Direct substitution of a variable from the parent for a parameter in the daughter.

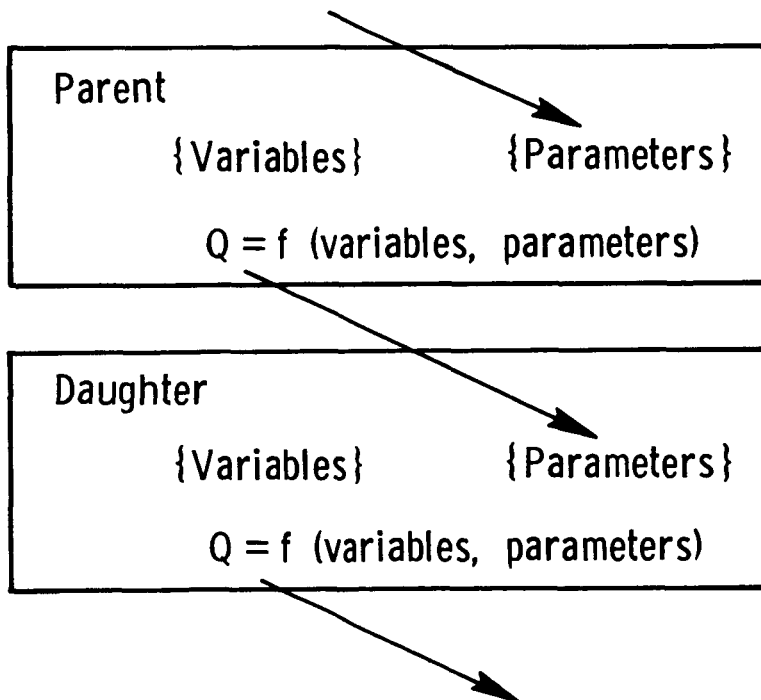


Fig. 4 A quantity computed in the parent substituted as a parameter in the daughter.

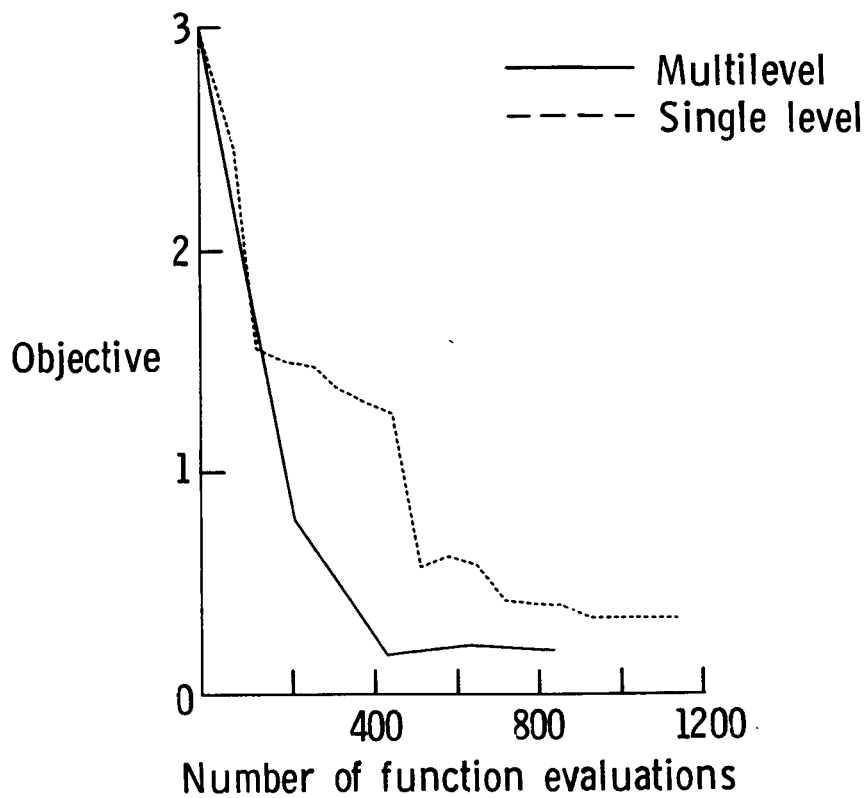


Fig. 5 Comparison of multilevel and single level histograms. Benchmark case with all design variables initialized to unity, $X = \{ 1. \}$.

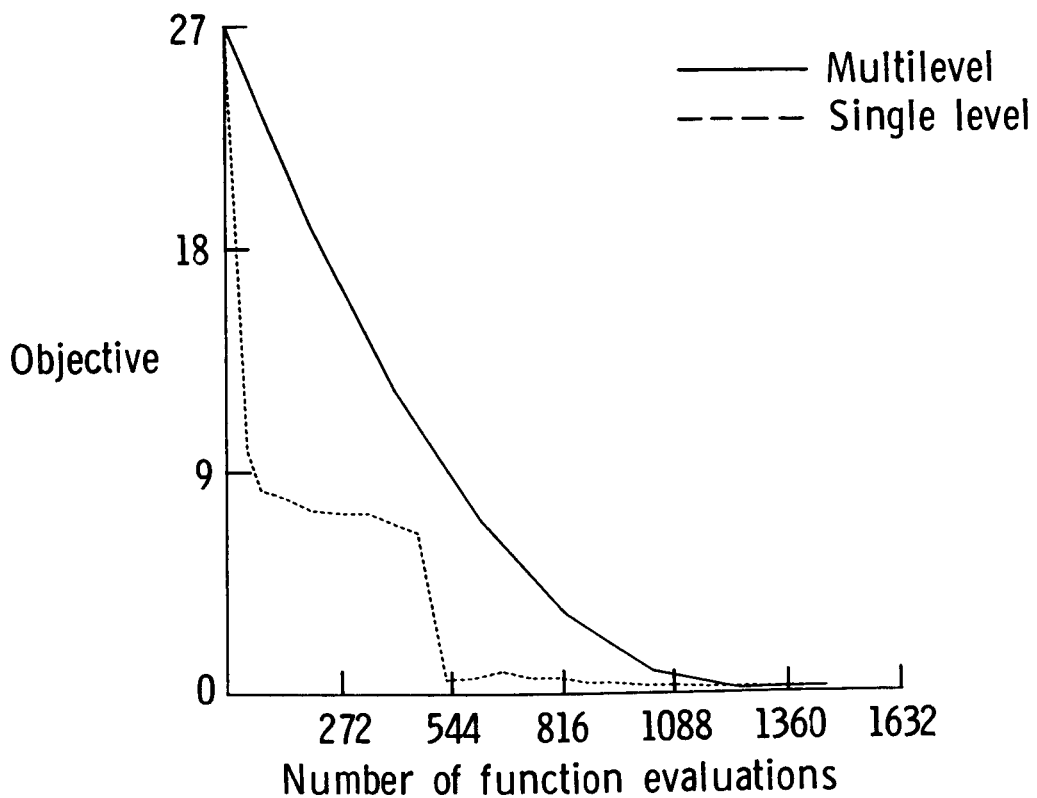


Fig. 6 Comparison of multilevel and single level histograms. Alternate initialization, $X = \{ 3. \}$.

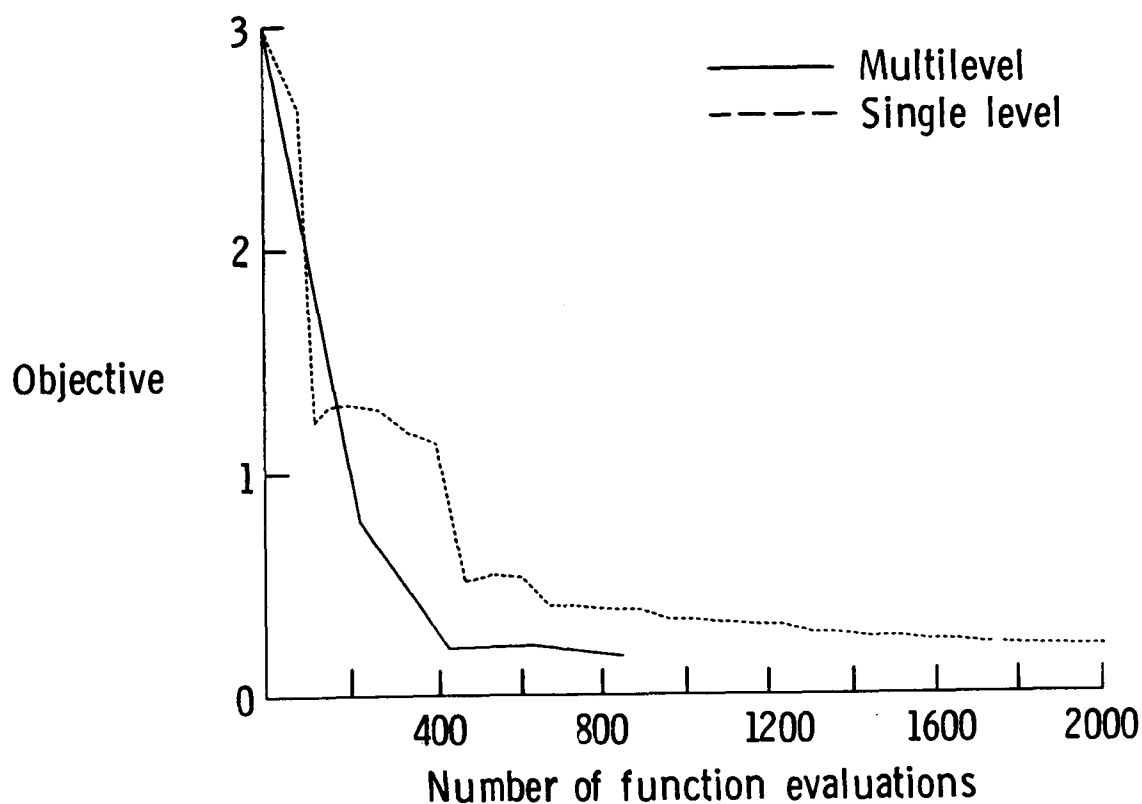


Fig. 7 Comparison of multilevel and single level histograms. Multiple parent case, $X = \{ 1. \}$.

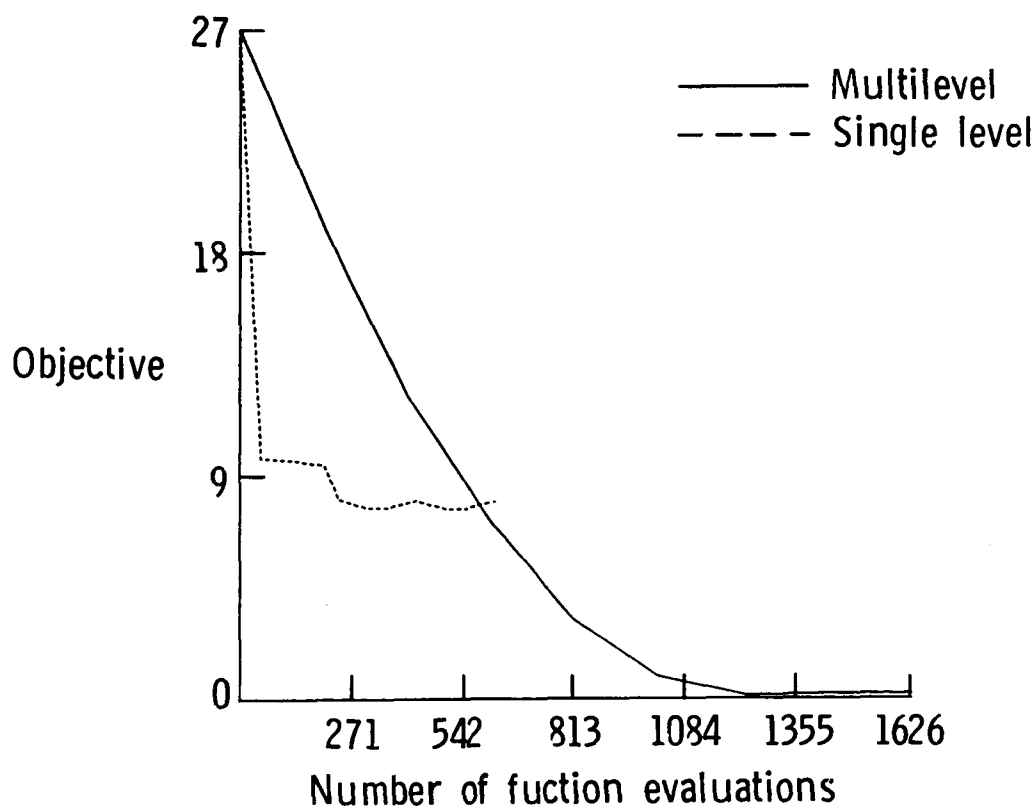


Fig. 8 Comparison of multilevel and single level histograms. Multiple parent case, $X = \{ 3. \}$.

Standard Bibliographic Page

1. Report No. NASA TM-89109		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A Computer Simulator for Development of Engineering System Design Methodologies				5. Report Date February 1987	
				6. Performing Organization Code 505-63-11-01	
7. Author(s) S. L. Padula and J. Sobieszczanski-Sobieski				8. Performing Organization Report No.	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665				10. Work Unit No.	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract A computer program designed to simulate and improve engineering system design methodology is described. The simulator mimics the qualitative behavior and data couplings occurring among the subsystems of a complex engineering system. It eliminates the engineering analyses in the subsystems by replacing them with judiciously chosen analytical functions. With the cost of analysis eliminated, the simulator is used for experimentation with a large variety of candidate algorithms for multilevel design optimization to choose the best ones for the actual application. Thus, the simulator serves as a development tool for multilevel design optimization strategy. The simulator concept, implementation, and status are described and illustrated with examples.					
17. Key Words (Suggested by Authors(s)) Systems engineering Optimization Computerized simulation			18. Distribution Statement Unclassified - Unlimited Subject Category - 59		
19. Security Classif.(of this report) Unclassified		20. Security Classif.(of this page) Unclassified		21. No. of Pages 16	
				22. Price A02	

For sale by the National Technical Information Service, Springfield, Virginia 22161